



Centro Universitario de Ciencias Exactas e Ingenierías.

Ciencias de la computación

Seminario de Uso, Adaptación y Explotación de Sistemas Operativos,
Becerra Velázquez Violeta del Rocío

Becerra Velázquez Violeta del Rocío

Hernandez Lomeli Diego Armando

219750396

INNI- Ingeniería en informática

D02

Actividad de aprendizaje 15

(3.2 Paralelismo)

2023/11/25

Tabla de contenido

| | |
|---|---|
| Seminario de Uso, Adaptación y Explotación de Sistemas Operativos, Becerra Velázquez Violeta del Rocío | 1 |
| Introducción: | 3 |
| Objetivo: | 3 |
| Instrucciones | 3 |
| Leer el documento Intel Parallel Studio | 3 |
| Realiza una investigación sobre las herramientas que componen a Intel Parallel Studio XE | 3 |
| a) Intel® Composer XE: Optimiza el rendimiento de los compiladores y las librerías. | 3 |
| b) Intel® Inspector XE: anteriormente Intel® Thread Checker, poderoso corrector de errores en hilos y memoria. | 3 |
| c) Intel® VTune™ Amplifier XE: anteriormente Intel® VTune™ e Intel® Thread Profiler, es una nueva herramienta de perfiles de rendimiento y se basa en el éxito de Intel® VTune™ Performance Analyzer. Incluye todas las funciones de Intel® Parallel Amplifier más un número de funciones avanzadas para desarrolladores que necesitan optimizar aún más el rendimiento. | 4 |
| d) Intel® Parallel Advisor: Herramienta auxiliar en la programación de hilos para desarrolladores en C/C++ Microsoft Visual Studio. | 4 |
| Una vez recolectada la información realice una síntesis exhaustiva donde exponga cada una de las herramientas que contiene Intel Paralell Studio y su relación con la programación multicore..... | 5 |
| Genere una conclusión sobre lo que ha aprendido en esta actividad | 8 |

Tabla de imágenes

| | |
|---|---|
| Ilustración 1 logotipo composer..... | 3 |
| Ilustración 2 logotipo "Inspector" | 3 |
| Ilustración 3 logotipo "Amplifier" | 4 |
| Ilustración 4 logotipo "Advisor" | 4 |
| Ilustración 5 pasos de diseño con "Parallel" | 6 |
| Ilustración 6 instrucciones ejecutados por procesador convencional, multiprocesador y tecnología Hyper-Treading (Roso, 2005)..... | 7 |
| Ilustración 7 representación de procesos e hilos | 8 |

Introducción:

Intel Parallel Studio XE es un conjunto completo de herramientas de desarrollo simplifica la creación y modernización de código con las últimas técnicas en vectorización, subprocesamiento múltiple, paralelización multinodo y optimización de memoria. Que permite crear código más rápido y aumentar el rendimiento de aplicaciones, simplificar códigos e implementar paralelismo en los programas.

La herramienta es tan extensa y poderosa que permite realizar un análisis detallado de los programas hechos en lenguajes de programación de alto nivel. Por lo que esta actividad presenta la herramienta como una opción a la optimización de aplicaciones a ejecutar en el sistema operativo sin entorpecer el trabajo de este.

Objetivo:

El estudiante relacionará los conocimientos previos con una herramienta que le permitirá optimizar sus programas y aplicaciones.

Instrucciones

Leer el documento Intel Parallel Studio

Realiza una investigación sobre las herramientas que componen a Intel Parallel Studio XE

- a) Intel® Composer XE: Optimiza el rendimiento de los compiladores y las librerías.

Ofrece un conjunto de herramientas que incluye:

- Intel®C++Compiler
- Intel®IntegratedPerformanceprimitives
- Intel®ThreadingBuildingBlocks
- Intel®MathKernelLibrary
- Intel®ParallelDebuggerExtension



Ilustración 1
logotipo composer

- b) Intel® Inspector XE: anteriormente Intel® Thread Checker, poderoso corrector de errores en hilos y memoria.

Herramienta de comprobación de errores que opera con memoria dinámica y permite el desarrollo en serie y multiprocesos dentro de Windows y Linux.



Ilustración 2
logotipo
"Inspector"

La capacidad de Inspector XE permite detectar los siguientes errores y los advierte:

- Errores de hilo
 - o Carrera de datos
 - o Interbloqueos
 - o Violaciones de jerarquía en los locks

Actividad de aprendizaje 15 (3.2 Paralelismo)

- Infracciones potenciales de seguridad
 - Información sobre los hilos
- Errores de memoria
 - Fugas de recursos
 - Llamadas a memoria no validas
 - Accesos a memoria no validaos
 - Cancelación de asignación o valida

- c) Intel® VTune™ Amplifier XE: anteriormente Intel® VTune™ e Intel® Thread Profiler, es una nueva herramienta de perfiles de rendimiento y se basa en el éxito de Intel® VTune™ Performance Analyzer. Incluye todas las funciones de Intel® Parallel Amplifier más un número de funciones avanzadas para desarrolladores que necesitan optimizar aún más el rendimiento.

Esta herramienta permite generar diferentes perfiles para así detectar cuellos de botella, su analisis se ejecuta a nivel algoritmo, con esto su enfoque va directamente en el código incluyendo la posibilidad de considerar el rendimiento según la microarquitectura del procesador.



Ilustración 3
logotipo
"Amplifier"

Los resultados del rendimiento son mostrados al usuario en forma de gráfica para comparar los resultados. El analisis avanzado hace un muestreo de eventos que mide que tan eficientemente se realiza elprocedimiento. También incluye detección de sobrecarga en paralelo.

Los algoritmos de analisis son:

- Hotspots
- Concurrencia
- Locks y esperas

Los resultados del analisis devuelven como resultado:

- Exploración general
- Accesos a memoria
- Ancho de banda
- Ciclos y uOpos
- Investigación de Front-End

- d) Intel® Parallel Advisor: Herramienta auxiliar en la programación de hilos para desarrolladores en C/C++ Microsoft Visual Studio.

Advisor como bien indica su nombre, es una guía que nos permite preparar código para la introducción al paralelismo, advisor proporciona secuencias de pasos que ayudan en la toma de decisiones para ejecutar paralelismo en el programa.



Ilustración 4
logotipo "Advisor"

Algunos de los beneficios que aporta son:

- Rendimiento: construye código optimizando los compiladores y eliminando cuellos de botella
- Confiabilidad: localiza errores en memoria e hilos
- Seguridad: detecta y elimina vulnerabilidades
- Flexibilidad: paralelismo
- Compatibilidad con Windows y Linux

Es posible usar advisor en editores de código como Visual Studio. Gracias a ello se pueden tomar decisiones para transformar un programa en serie a uno con paralelización. Asimismo, se debe generar un proyecto que pueda proporcionar un ejecutable para cada herramienta de advisor.

Una vez recolectada la información realice una síntesis exhaustiva donde exponga cada una de las herramientas que contiene Intel Paralell Studio y su relación con la programación multicore.

Además de las herramientas anteriores existe **Static Security Analysis (SSA)** su utilidad va a detectar vulnerabilidades de las aplicaciones y en detección de errores de programación.

Sus resultados muestran los siguientes problemas:

- Desbordamiento de buffer
- Mal uso de punteros y de almacenamiento
- Mal uso del lenguaje C/C++ o sus librerías
- Mal uso del lenguaje Fortran o sus librerías
- Mal uso de OpenMP
- Mal uso de Cilk Plus

Además, sus resultados de análisis no generan ejecutables, sino archivos con los resultados del análisis.

La utilización de las herramientas de **Parallels Studio** NO GARANTIZAN que el programa generado sea una versión mejorada del programa en serie original aunque se sigan correctamente los pasos, metodologías de desarrollo y guías de parallels.

Su metodología se puede resumir en 4 pasos generales

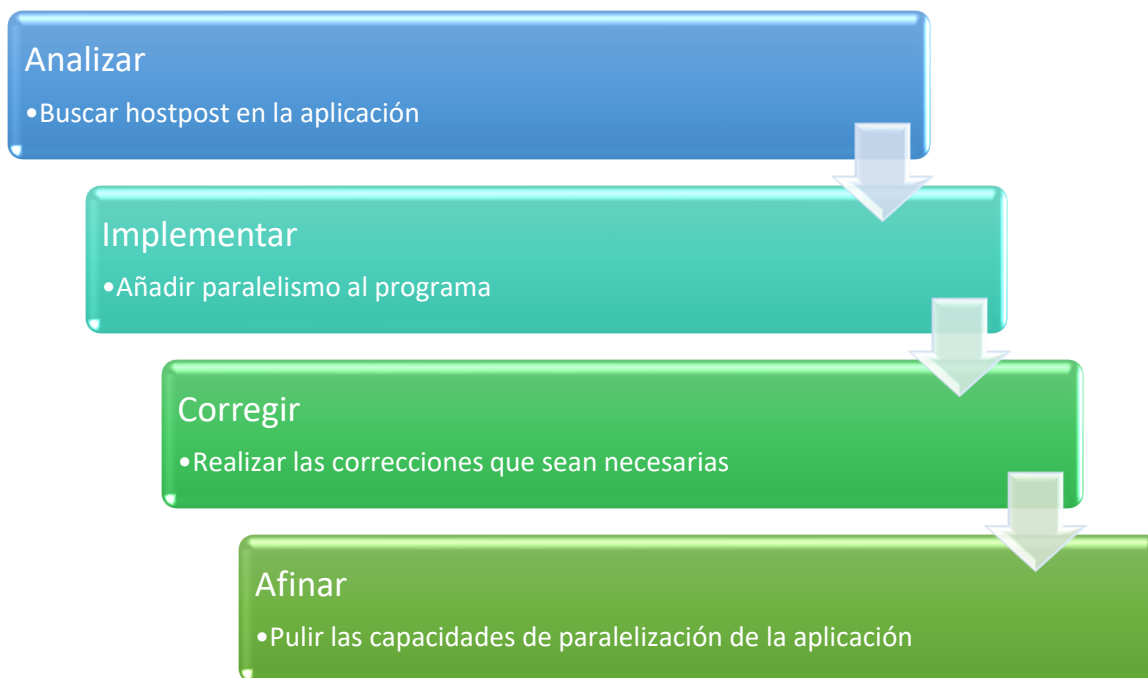


Ilustración 5 pasos de diseño con "Parallel"

Sabemos hasta este punto que **Paralells** es útil para mejorar implementaciones que requieran del paralelismo para su resolución o posible mejora, pero ¿En que consiste realmente?

Esta parte de la programación consiste en el uso de los diferentes núcleos e hilos que un sistema computacional utiliza para completar las tareas que necesite, los microprocesadores con esta capacidad se realizan con arquitecturas de memoria compartida. Algunas arquitecturas conocidas son **SIMD** y **MIMD** ambos son arquitecturas que permiten la paralelización de maneras diferentes.

También existe el **GPU Computing**, que incrementa el rendimiento de las aplicaciones, el uso del **GPU** permite desahogar al procesador de procesos de gran capacidad computacional, es un trabajo en conjunto con el **CPU** que ejecuta el código de un programa mientras el **GPU** se encarga de actividades de alta exigencia técnica.

A esto se le conoce como “coprocesamiento” que en aspectos técnicos son superiores al “procesamiento central”, la tendencia a dividir el procesamiento en varias secciones fue generada por la necesidad de mejorar el rendimiento, la elección de esta estrategia permite a los procesadores multicore (multinúcleo) atender procesos sin necesidad de tener que esperar la resolución de otros para poder continuar, gracias a esto, mientras se ejecuta una función, otro corea tendera otra, evitando tiempos de espera y responder múltiples actividades a la vez.

Sin embargo, más no siempre es mejor, a mayor número de núcleos, debe existir la capacidad de acceso a memoria suficiente para atender los accesos constantes, este problema es generado por la estructura de los procesadores y termina en lo que se conoce como cuello de botella, esta situación provoca que se desperdicie la capacidad de los componentes dentro de un equipo, en el contexto del procesamiento, se puede mermar el uso de los procesadores por la espera activa de

acceso a la memoria compartida o en caso contrario, los procesadores podrían no llevar al máximo la capacidad de lectura de la memoria, con eso se desperdiciaría la capacidad de lectura.

Debemos considerar los costos de producción, no es solo el diseño e implementación del hardware, también se debe contar con la capacidad lógica para aprovechar el hardware (software), una posible alternativa es la aplicación del **hyper Treading**, dicho método hace que un microprocesador trabaje como 2 CPU's virtuales, gracias a esto se podrían ejecutar varios hilos de ejecución en paralelo.

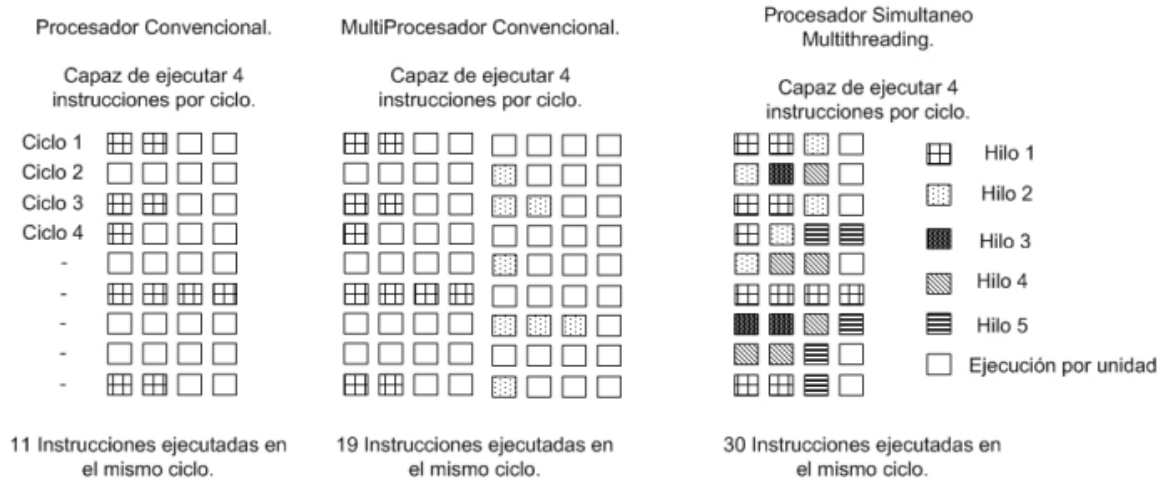


Ilustración 6 instrucciones ejecutadas por procesador convencional, multiprocesador y tecnología Hyper-Treading (Roso, 2005)

Una alternativa más moderna es el multicore, permite a un solo equipo tener más de 1 procesador físico, cada uno es independiente, contando con su propia memoria L2 de 1 o 2 Mb de capacidad, la capacidad de procesadores depende del tipo principal del procesador.

Con la implementación del multicore, los hilos son aun más presentes en la programación, también conocido como programación paralela y concurrente, su principal propósito es facilitar la sincronización entre proceso y aumentar la eficacia en alternancia de procesos. Un único proceso puede contener diferentes hilos de ejecución para su propia conveniencia, a esto se le conoce como multihilo, el sistema también debe ser capaz de mantener la ejecución de estos hilos.

Tradicionalmente se podía ver a los sistemas operativos como procesamiento monohilo, haciendo 1 cosa a la vez, pero en un entorno multihilo a un proceso se le define como la unidad de protección y asignación de recursos como se muestra a continuación:

- Espacio de direcciones virtuales en la imagen del proceso.
- Acceso protegido a los procesadores E/S para concurrencia
- Los procesos pueden generar hilos que a su vez también pueden generar más hilos.
- El contexto del hilo se guarda en el procesador cuando no está siendo ejecutado (uso de memoria cache)
- Pila de ejecución

Actividad de aprendizaje 15 (3.2 Paralelismo)

- Almacenamiento estático para variables locales
- Acceso a la memoria y recursos del proceso

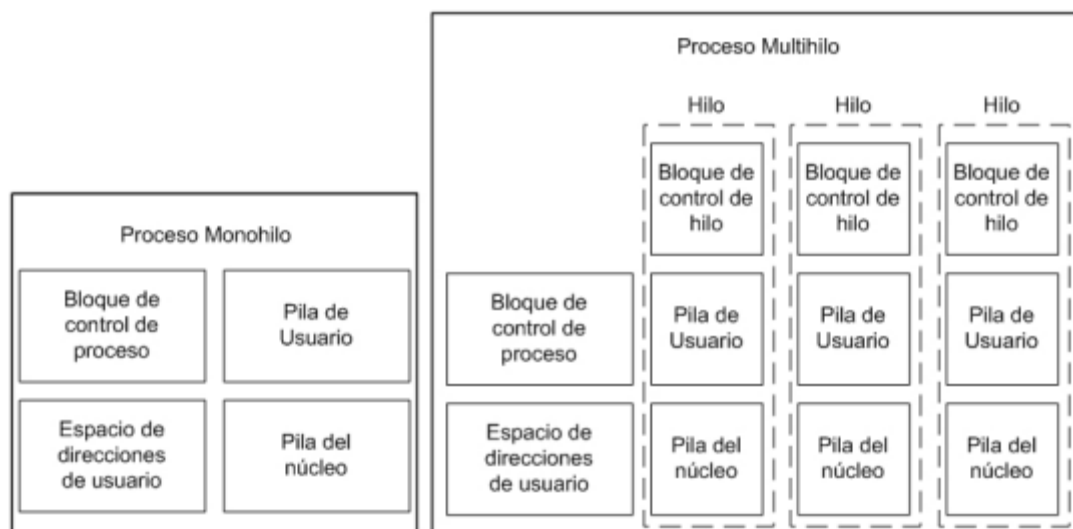


Ilustración 7 representación de procesos e hilos

La implementación multicore también tiene sus propias desventajas, hablamos de inancion, interbloqueo, tiempos de espera activa, etc. Y dichos problemas han de ser enfrentados y superados, pero ¿cómo es que se pueden resolver problemas tan complejos? Existen diferentes marcos de trabajo y metodologías pero es necesario el uso de herramientas especializadas que permitan enfocar de manera inteligente los recursos y trabajo invertidos.

Intel Parallel Studio XE es un software creado y dirigido por Intel para el desarrollo de programas paralelos, gracias a este paquete de herramientas se pueden corregir errores y facilitar el desarrollo con mejoras en el rendimiento, además, la calidad del código aumenta junto a la seguridad y fiabilidad para sistemas empresariales.

Se mantiene como objetivo con **Parallel Studio XE** simplificar la migración de procesadores multicore a Many-core.

Genere una conclusión sobre lo que ha aprendido en esta actividad

Las tendencias son situaciones de impacto sobre un tema en específico y que además permiten evolucionar ese mismo contexto, dentro de la programación, cuando se comenzó a popularizar el uso del procesamiento en paralelo, era inevitable que surgieran nuevas herramientas que permitieran obtener el mayor potencial posible en cada nuevo desarrollo posible y dentro de las mejoras comienzan a nacer facilidades para satisfacer la demanda de dicha tendencia, Parallel Studio es un ejemplo de como el surgimiento de nuevas estrategias y tecnologías pueden seguir evolucionando y hacerse más poderosas con el paso del tiempo.

Bibliografía

Multicore. (s.f.). En *Arquitectura de Computadoras Avanzada* (págs. 247 - 304).